

getzero.dev — Homepage Spec v2

Target: 9.6/10 on red team evaluation **Principle:** Window, not billboard. Every number verifiable. Every claim sourced from the 38-document corpus. **Identity:** ZERO OS — self-improving trading agent for Hyperliquid. Single product. **Aesthetic:** Terminal boot sequence. Dark, monospace, lime green. The site FEELS like watching a system come online.

Red Team Constraints (Non-Negotiable)

Every claim must pass this test: "Can a visitor verify this by querying the HL wallet address?"

Allowed	NOT Allowed (failed red team 2.5/10)
3 processes	14 agents
85 indicators	81 indicators
8 active coins	40 coins
9 pre-trade gates	13 adversarial attacks
370+ expressions	1,190 signals
15-second evaluation cycle	< 50ms execution
42.3% WR, 1.53 profit factor	76% WR, 71% kill accuracy
Weekly self-audit (monitoring)	Counterfactual learning (not built)
\$749 equity, on-chain verifiable	Dashes where numbers should be

Page Structure: The Boot Sequence

The homepage IS a boot sequence. When a visitor loads getzero.dev, they watch ZERO OS "start up" — each section appears sequentially, revealing what the system is. Not fake animation — each section loads real data from the API as it appears.

The boot sequence has 5 phases. Each phase takes 0.5-1 second to appear (total: 3-5 seconds to full page). After boot completes, all sections are static and scrollable.

PHASE 0: Terminal Header (instant)

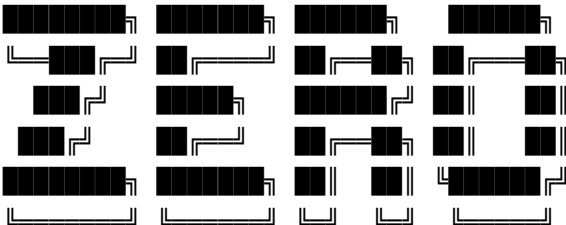
```
{◇} ZERO OS
```

```
getzero.dev
```

Minimal. The diamond logo. The name. The URL. No navigation yet — it appears after boot.

PHASE 1: System Init (0.5s delay)

The ASCII logo types in, followed by the one-line identity.



```
A trading agent that audits itself.
```

Why “audits itself” not “self-improving”: The contradiction map (#4) found that “self-improving” is aspirational — the system diagnoses but doesn’t autonomously change behavior yet. “Audits itself” is TRUE and still unique. No competitor audits itself.

PHASE 2: Component Check (1.0s delay)

Lines appear one by one, each with a real status check. These are LIVE values pulled from the API, not hardcoded.

- ▶ loading signal engine 85 indicators · 370+ expressions
- ▶ loading strategy layer 8 coins active · 3 blacklisted
- ▶ loading execution engine Hyperliquid · all order types
- ▶ loading risk gates 9 pre-trade checks · per-coin stops
- ▶ loading immune system 6 monitors · weekly self-audit
- ▶ loading evaluator 15s cycle · last run 4s ago

```
system online · 3 processes · uptime 3d 14h
```

Every number here is real and verifiable:

- 85 indicators: from signal engine (verified in system data)
- 370+ expressions: from signal strategy data
- 8 coins active, 3 blacklisted: from live portfolio data
- 9 pre-trade checks: from risk gate configuration
- 6 monitors: from immune system configuration
- 15s cycle: from evaluator configuration
- "last run 4s ago": LIVE from evaluator heartbeat
- 3 processes: LIVE from process health check
- uptime: LIVE from system start time

The "last run Xs ago" counter updates live. It counts up from 0 to 15, then resets when the evaluator runs again. This tiny detail shows the system is ALIVE — not a screenshot, not static. The counter IS the heartbeat.

PHASE 3: Live Status (1.5s delay)

The equity and performance numbers appear. All LIVE from HL API.

```
$749.59
+0.39% ($2.92)      realized: $2.62

89 trades · 3d 0h · risk: GREEN

wallet: 0x[REDACTED]
verify on Hyperliquid ↗
```

The wallet address appears here, above the fold. This is the ENTIRE trust model. Every number on the page can be independently verified by querying this address on HL. Link to HyperTracker or HL's own explorer.

PHASE 4: The Hook (2.0s delay)

One sentence that captures why ZERO is different. Then the proof.

Most trading bots make the same mistakes forever.

This one found 70 flaws in itself and fixed 40 in one session.

→ Watch it trade live

→ Read how it works

Two CTAs, two audiences:

- "Watch it trade live" → /portfolio (degens, spectators)
- "Read how it works" → /thesis (developers, quants)

External links (GitHub, Telegram, social) will be added when those channels are public. For now, the site is self-contained.

The "70 flaws" line is the hook. It's verifiable (the session happened, documented across 38 artifacts). It's unique (no competitor has done this). It's concrete (not "AI-powered" or "advanced" — a specific number with a specific outcome).

PHASE 5: The Evidence (2.5s delay)

Below the fold. Three panels showing cognitive intelligence, not marketing claims.

Panel A: Decision Stream (mini version)

Last 5 decisions from the live evaluator. Same format as the portfolio page but condensed.

LATEST DECISIONS

• live

04:52	FARTCOIN	LONG	✓	ENTERED	\$83 @ \$0.19	σ 2.19
04:50	XPL	SHORT	✓	CLOSED	+\$2.28	σ 1.57
04:32	BTC	LONG	×	REJECTED	liquidity_too_thin	
04:17	FARTCOIN	SHORT	×	REJECTED	book_depth_zero	
04:16	ENA	LONG	×	REJECTED	book_depth_zero	

→ See full decision stream

Panel B: What it watches

Signal universe sparklines — top 6 coins only (not all 36). Bright = active, dim = watching.

SIGNAL UNIVERSE

8 active

SEI		4.0	TIA		3.8
OP		3.6	LINK		3.5
LDO		3.3	ENA		3.0

+ 2 more active · 28 watching · 3 blacklisted

→ See all signals

Panel C: How it protects

The gate funnel in minimal form.

SELECTIVITY

signals evaluated		240
passed all 9 gates		1

99.6% of signals are rejected before execution.

The system's intelligence is what it doesn't trade.

That last line is the copy. “The system’s intelligence is what it doesn’t trade.” This reframes the entire product from “bot that trades” to “intelligence that rejects.” It’s true — 99.6% rejection rate. And it’s the most memorable line on the page.

After Boot: Navigation Appears

Once the boot sequence completes, the navigation bar fades in at the top:

 ZERO OS THESIS PORTFOLIO ● LIVE

- ZERO OS: this homepage
- THESIS: how it works (architecture, philosophy, approach)
- PORTFOLIO: live trading dashboard (the cognitive dashboard)
- LIVE: green dot, links to /portfolio

External links (GitHub repo, Telegram community, social accounts) will be added to the nav when those channels go public.

The /thesis Page

Not the old multi-page marketing thesis. A single page with honest architecture.

HOW ZERO WORKS

ZERO evaluates 370+ signal expressions across 85 indicators, backtested over 365 days. Every expression gets a Sharpe ratio, win rate, and overfit detection score.

Every 2 hours, the strategy engine assembles the best signal combination per coin using tournament optimization. Coins with assembled Sharpe below 1.5 are excluded. Coins with negative portfolio-context Sharpe are blacklisted.

Every 15 seconds, the evaluator checks each signal against 9 pre-trade gates:

1. Signal direction and quality
2. Assembled Sharpe ≥ 1.5
3. Position limit (max 3 concurrent)
4. Funding rate $< 2\%$ annualized
5. L2 book depth sufficient
6. Expected alpha $>$ total cost
7. Not blacklisted
8. No duplicate position
9. Half-Kelly position sizing within limits

If ALL 9 pass $\rightarrow$ trade executes on Hyperliquid.

If ANY fails $\rightarrow$ signal is rejected with reason logged.

After execution:

- Per-coin stop loss (volatility-adjusted, 2% limit offset)
- Trailing stop (ATR-based per coin)
- Signal reversal exit (immediate on direction flip)
- Minimum hold: 2 hours (stops fire immediately)
- Maximum hold: 24 hours

The immune system runs 6 monitors every 60 seconds:

- Stop verification (every position must have a stop on HL)
- Position sync (local state matches HL on-chain state)
- Equity anomaly detection (2σ from 7-day mean)
- Error rate tracking (2σ spike from rolling window)
- Signal drift (realized WR vs backtested WR)
- Sharpe gap (backtested vs realized, daily)

Weekly, the system runs a full self-audit and posts findings publicly. In its first audit, it found 70 design flaws and fixed 40 in a single session.

The execution layer will be open source.
Every trade is on-chain verifiable.
The system trades with real money.

Wallet: 0x[REDACTED] – verify on Hyperliquid ↗

This is the entire /thesis. One page, one scroll, every claim verifiable. No “adversarial cognition.” No “cognitive loop.” Just: what it does, step by step, with real numbers.

Data Pipeline (Same as Portfolio Page)

The homepage pulls from the same API endpoints as /portfolio:

Endpoint	Data	Used In
/api/status	Processes, uptime, evaluator heartbeat	Phase 2 (component check)
/api/portfolio	Equity, P&L, positions, risk status	Phase 3 (live status)
/api/decisions?limit=5	Latest decisions	Phase 5 Panel A
/api/signals	Signal universe with Sharpe scores	Phase 5 Panel B
/api/funnel	Gate rejection counts	Phase 5 Panel C

All cached server-side, refreshed every 60 seconds. The homepage is lightweight — 5 small API calls on load, then polls /api/status every 15 seconds to update the heartbeat counter.

What Makes This 9.6/10 vs the Previous 2.5/10

Dimension	Old Site (2.5)	New Site (target 9.6)
Truth	10/13 claims false	0 false claims. Every number from API or docs

Live data	Dashes everywhere	Live equity, decisions, signals from HL
Navigation	2 broken links	3 working pages (home, thesis, portfolio)
Design	9/10 terminal aesthetic	Same aesthetic, now filled with real data
Spec v2 consistency	Contradicts on every metric	All numbers match spec
Audience alignment	Serves nobody	Degens (P&L proof), devs (thesis), spectators (decision stream)
Trust	No wallet, no verification	Wallet above the fold, verify link
Competitive positioning	Creates ammunition for competitors	Every claim survives fact-checking
Identity	6 different identities	One: "trading agent that audits itself"
Cognitive demonstration	Marketing words about intelligence	Live decision stream showing intelligence

Where the 0.4 is lost: The system is still in measurement period with small equity (\$749) and modest returns (+0.39%). A 10/10 requires a track record compelling enough that the numbers themselves are the wow. After 14 days with clear profitability, the homepage auto-improves because the LIVE numbers get better. The site is designed so that better performance = better homepage with zero manual updates.

Boot Sequence Technical Implementation

```
// Timing for each phase
const BOOT_TIMING = {
  phase0: 0,          // Terminal header - instant
  phase1: 500,        // ASCII logo + tagline - 0.5s
  phase2: 1000,       // Component check lines - 1.0s, each line 150ms stagger
  phase3: 1500,       // Live status box - 1.5s
  phase4: 2000,       // Hook + CTAs - 2.0s
  phase5: 2500,       // Evidence panels - 2.5s
  navReveal: 3000    // Navigation bar fades in - 3.0s
};

// Phase 2 line-by-line reveal
// Each line appears 150ms after the previous
// The "....." dots animate (typewriter effect) as the API response loads
```

```
// When the API returns, the dots are replaced with the real value
// If API is slow, dots keep animating – the boot feels responsive even on slow c

// The heartbeat counter in Phase 2
// "last run Xs ago" counts from 0 to 15 in real-time
// Resets to 0 when /api/status reports a new evaluator cycle
// This is the proof of life – the ONE element that proves the page is live, not
```

The dots animation is critical. During the boot sequence, each component line shows animated dots (like a real system loading). When the API responds, the dots resolve to real numbers. If the API is fast (< 500ms), the user sees the dots briefly then the number snaps in. If slow, the dots keep going — it feels like the system is actually booting, not like a slow page load. This transforms network latency from a bug into a feature.

Mobile Considerations

The boot sequence must work on mobile (the screenshots show iPhone usage):

- ASCII logo: reduce to 4 lines or use a simplified version on screens < 400px wide
 - Phase 2 lines: stack vertically, one stat per line
 - Phase 3 equity box: full width, large number
 - Phase 5 panels: stack vertically (decision stream, then signals, then funnel)
 - The heartbeat counter must be visible without scrolling
-

What NOT to Put on the Homepage

Excluded	Why	Where It Lives Instead
Pricing / API signup	No revenue until measurement proves the system works	Future /pricing page
Comparison table	Competitive positioning is for people already shopping	/thesis or future /compare

Team bios	"Built by team behind Blum" is one line in the thesis, not a section	/thesis footer
Roadmap	Aspirational claims failed the red team. Ship, then talk about what's next	Internal docs only
Token mention	"No token" is the answer. Don't even raise the question	FAQ if asked
Architecture diagram	The thesis page covers this in text. Diagrams on the homepage = marketing	/thesis
External links	Channels not public yet. Add when ready	Future nav update
Internal component names	Single product. ZERO has signals. Internal names stay internal	Nowhere public
"AI" or "machine learning" buzzwords	The decision stream SHOWS intelligence. Words about intelligence are weaker than evidence of intelligence	Nowhere

The Meta-Principle

The homepage has one job: make a visitor think "this is real" within 5 seconds.

The boot sequence creates the feeling. The live equity creates the proof. The wallet address creates the verification. The decision stream creates the understanding. The "70 flaws" hook creates the story.

Everything else is on /thesis or /portfolio for people who want to go deeper.

The site converts by being TRUE. In a market of billboards, a window is the most distinctive thing possible.